A systemd primer

Pat Riehecky
Scientific Linux, Fermilab

July 2014

Getting a few things out of the way

- Lots of people feel strongly about systemd
- Systemd in SL
  - It is the default init system for SL7
  - It is not available for SL5 or SL6
  - This follows upstream

A Conceptual Shift

Systemd is a system/service manager
Sysvinit is a 'script runner'

Understanding this difference will help you understand its behavior

A Conceptual Shift

Systemd is an application and toolkit
Sysvinit is neither

Understanding this difference will help you work with it, rather than against it.

A Conceptual Shift

Systemd knows about your system
Sysvinit doesn't want to

Understanding this difference will help you understand its benefits

A Conceptual Shift

Systemd has systemctl
Sysvinit has chkconfig and service

```
service sshd start == systemctl start sshd.service
chkconfig sshd on == systemctl enable sshd.service
```

A Conceptual Shift

- Sysvinit
  - Run 'start' script on runlevel
  - Run 'stop' script on runlevel

- Systemd
  - Run 'start' command if needed:
    - Required for other service
    - Required for 'runlevel'
    - Abnormal termination
    - Dbus trigger
  - Run 'stop' command if needed:
    - Required for 'runlevel'
    - Dbus trigger

A Conceptual Shift

- Sysvinit
  - Output
    - STDOUT
    - STDERR
    - Maybe syslog too
  - Application errors
    - Maybe /var/log/messages

- Systemd
  - Output
    - Systemd-journal
    - And to syslog too
  - Application errors
    - journalctl -u sshd.service
    - /var/log/mesages

Systemd Benefits

- Service tracking
- Dependency tracking
- Socket tracking
- Automated respawns
- Unified reporting
- Automatic CGroup assignment
- A fully exposed API
- Simple service definitions
- Simple parameter overrides
- Users can make and manage their own services

A systemd unit file: sshd.service

```
[Unit]
Description=OpenSSH server daemon
After=syslog.target network.target auditd.service
Documentation=man:sshd(8)

[Service]
EnvironmentFile=/etc/sysconfig/sshd
ExecStartPre=/usr/sbin/sshd-keygen
ExecStart=/usr/sbin/sshd -D $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s

[Install]
WantedBy=multi-user.target
```

This does everything you found in /etc/init.d/sshd and adds all the systemd features

A systemd unit file (part 1)

```
[Unit]
Description=OpenSSH server daemon
After=syslog.target network.target auditd.service
Documentation=man:sshd(8)
```

- The [unit] section describes the item as a whole
- 'After' is a list of things which must be completed before this can be started
- 'Documentation' is provided to the admin/user via
  - systemctl help sshd.service

A systemd unit file (part 2)

```
[Service]
EnvironmentFile=/etc/sysconfig/sshd
ExecStartPre=/usr/sbin/sshd-keygen
ExecStart=/usr/sbin/sshd -D $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s
```

- The [service] section describes the attributes of the service
- EnvironmentFile is imported into the runtime environment of the service
- ExecStartPre is run before ExecStart and should return 0 before ExecStart is run
- ExecStart is run to start the service
- ExecReload is run to reload the service's configuration
- KillMode describes how to kill this service's processes
- Restart will run ExecStart if the process exits before being stopped by systemd
- RestartSec is the delay between when the process ends and when it is restarted

A systemd unit file (part 3)

```
[Install]

WantedBy=multi-user.target
```

- The [Install] section describes the behavior of the service, relating to how it is loaded by systemd
- WantedBy lists systemd elements that are 'incomplete' until they've reviewed this unit.
  - So, multi-user.target is 'incomplete' until it has checked to see if sshd.service is enabled or disabled and acted upon that information.

Adding/Changing systemd units

Systemd units are defined or inherit settings in a clear work flow

1) /usr/lib/systemd/system/*
2) /run/systemd/system/*
3) /etc/systemd/system/*

So, a service defined at (1) can have its values altered at (2) and those values can be further altered at (3) with the end result being any unchanged values from a lower entry kept by the higher one.

What?

Changing systemd units: an example

/usr/lib/systemd/service/mariadb.service

```
[Unit]
Description=MariaDB database server
After=syslog.target network.target

[Service]
Type=simple
User=mysql
Group=mysql

ExecStart=/usr/bin/mysqld_safe
PrivateTmp=true


[Install]
WantedBy=multi-user.target
```

/etc/systemd/system/mariadb.service

```
[service]
User=mariadb
Group=nobody
LimitNOFILE=1000
```

The files in /usr/lib come from the RPM and will be replaced.

The files in /etc/systemd come from you and will NOT be replaced

Socket Management

- inetd/xinetd are gone
  - The only thing they provided apart from sysvinit was a network socket
- Systemd now provides this via '.socket' units
  - Each .socket unit requires a .service unit too

Rsyncd.service

```
[Unit]
Description=remote file copy program daemon
ConditionPathExists=/etc/rsyncd.conf

[Service]
EnvironmentFile=/etc/sysconfig/rsyncd
ExecStart=/usr/bin/rsync --daemon "$OPTIONS"

[Install]
WantedBy=multi-user.target
```

Rsyncd.socket

```
[Unit]
Description=Rsync Server Socket
Conflicts=rsyncd.service

[Socket]
ListenStream=873
Accept=yes

[Install]
WantedBy=sockets.target
```

Socket Management

To start the rsyncd socket simply run:

```
systemctl start rsyncd.socket
```

To start the rsyncd socket on boot run:

```
systemctl enable rsyncd.socket
```

Compare to ssh:

```
systemctl start sshd.service
systemctl enable sshd.service
```

Non-root service management

- Non-root users can run services
  - This has always been true.
    - Just pick a port above 1024 and bind.
- Systemd makes this easy
  - Uses same config syntax as normal, but different locations
    - /usr/lib/systemd/user/*
    - /run/systemd/user/*
    - /etc/systemd/user/*
    - $HOME/.config/systemd/user/*
    - $XDG_CONFIG_HOME/systemd/user/*
  - Simply set systemctl to 'user' mode
    - systemctl --user start someuser-vncserver.service

Reviewing log files

- On SL6, you find the log files and read them
- On SL7, you can simply ask:
  - journalctl -u sshd.service
  - Want the output in .json format, just add -o json
  - journalctl --since yesterday -u sshd.service
  - journalctl --until 2014-07-29
  - Want to read the log files, they are still there right where you'd expect

Get A Quick Status

- SL6

```
# service sshd status
openssh-daemon (pid  2715) is running...
```

Get A Quick Status

- SL7

```
# systemctl status sshd.service
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
   Active: active (running) since Mon 2014-05-12 18:01:24 CDT; 2 months 11 days ago
     Docs: man:sshd(8)
 Main PID: 1425 (sshd)
   CGroup: /system.slice/sshd.service
           └─1425 /usr/sbin/sshd -D

Jul 23 17:36:39 test.fnal.gov sshd[17902]: Authorized to root, krb5 principal csieh@FNAL.GOV (krb5_kuserok)
Jul 23 17:36:39 test.fnal.gov sshd[17902]: Accepted gssapi-with-mic for root from 131.225.122.34 port 58024 ssh2
```

Get A Quick Status

- SL6

```
# chkconfig --list sshd
sshd           0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

Get A Quick Status

- SL7

```
# systemctl is-enabled sshd.service
enabled
# echo $?
0
```

For more information

- http://www.freedesktop.org/wiki/Software/systemd/
  - Especially the 'systemd for Administrators' series
- RHEL7 System's Administrators Guide
  - Section III. Infrastructure Services
- https://access.redhat.com/videos/403833
- https://access.redhat.com/videos/898503
- https://access.redhat.com/videos/898473
-